A Not So Short iSCSI Tutorial

Ming Zhang (mingz@ele.uri.edu)
HPCL, University of Rhode Island
10/03

Outline

- Reexamine SCSI and SAN.
- Why do we need iSCSI?
- What is iSCSI?
- Go deeper - examine the iSCSI protocol.
- Live example – UNH iSCSI reference implementation
- Nothing is perfect: iSCSI limitation.

SCSI Protocol

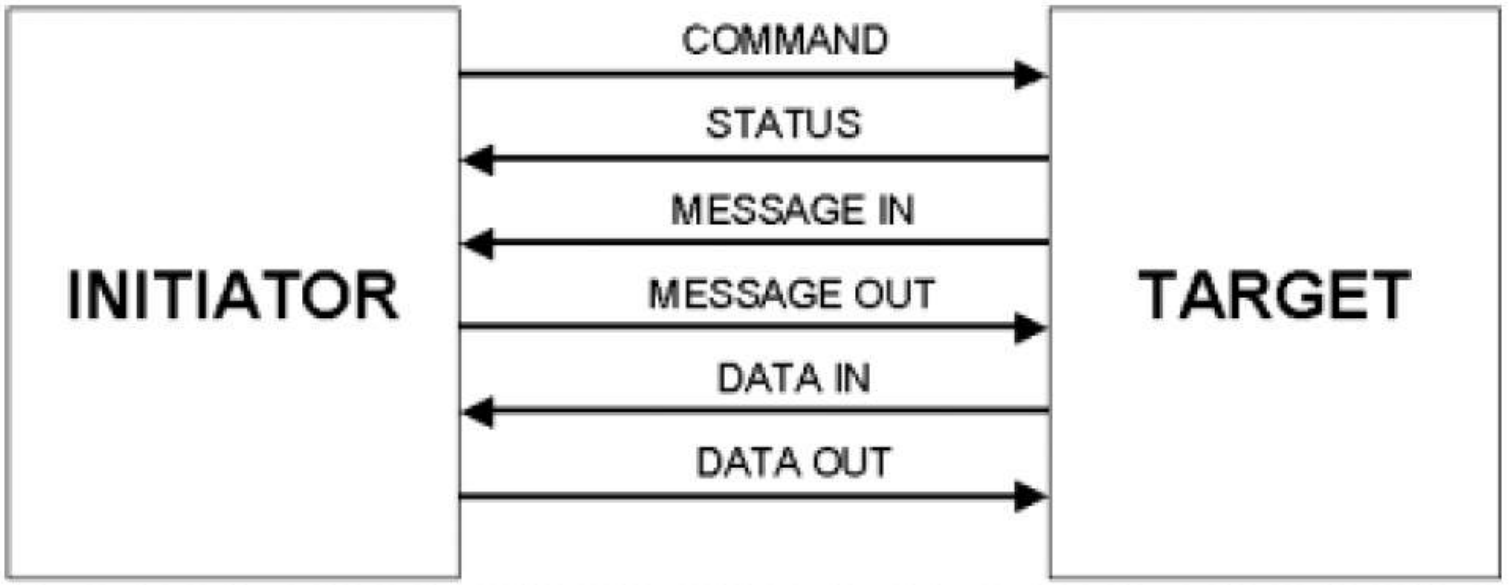
- SCSI - Small Computer System Interface;
- SCSI became an ANSI standard in 1986;
- SCSI is both a bus hardware specification and a command set;
- SCSI makes computer systems complete device independent.
- SCSI has high speed range from 1MB/s – 320MB/s

Common words in SCSI

- SCSI **Initiators** are the SCSI devices that start the I/O process and SCSI **Targets** are the SCSI devices that respond to a request to perform an I/O process.
- CDB - Command Descriptor Block
- LUN - Logical Units (max 8 LUNs per device)
- LBA - Logical Block Address
  - A linear address space mode to mask the details.

8 SCSI Command Phases

- BUS FREE, ARBITRATION, SELECTION, RESELECTION,
- Information Transfer Phases:
  - COMMAND, DATA, STATUS, MESSAGE;



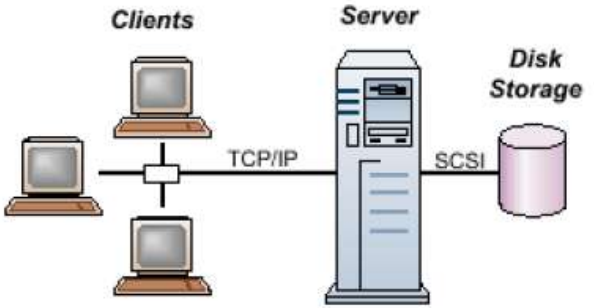
SCSI Commands

- Several important SCSI commands:
  - TEST UNIT READY,
  - READ CAPACITY,
  - READ 6, 10, 12,
  - WRITE 6, 10, 12,
  - REQUEST SENSE.

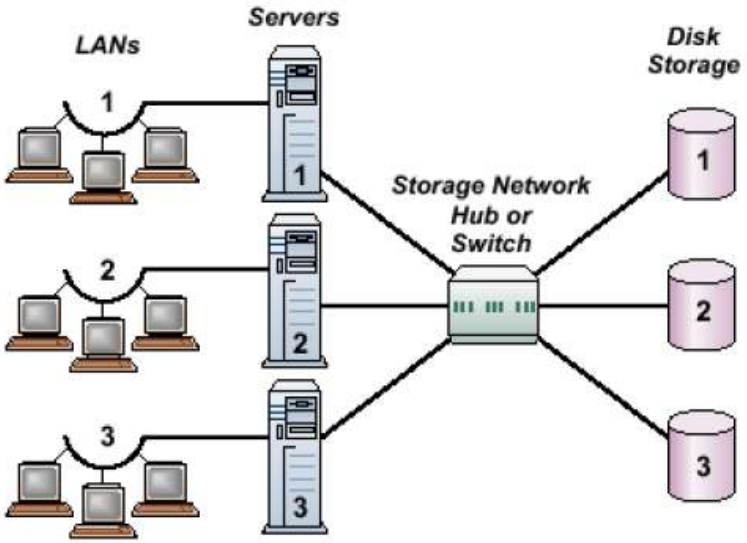
Bit Byte	7	6	5	4	3	2	1	0
0	Operation Code (08h)							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address							
4	Transfer Length							
5	Vendor Unique		Reserved			NACA	Obsolete	Link

SAN vs. (Direct-attached) SCSI

- Any to any;
- Consolidation;
- Availability
- Scalability
- Bandwidth



Sample LAN Configuration



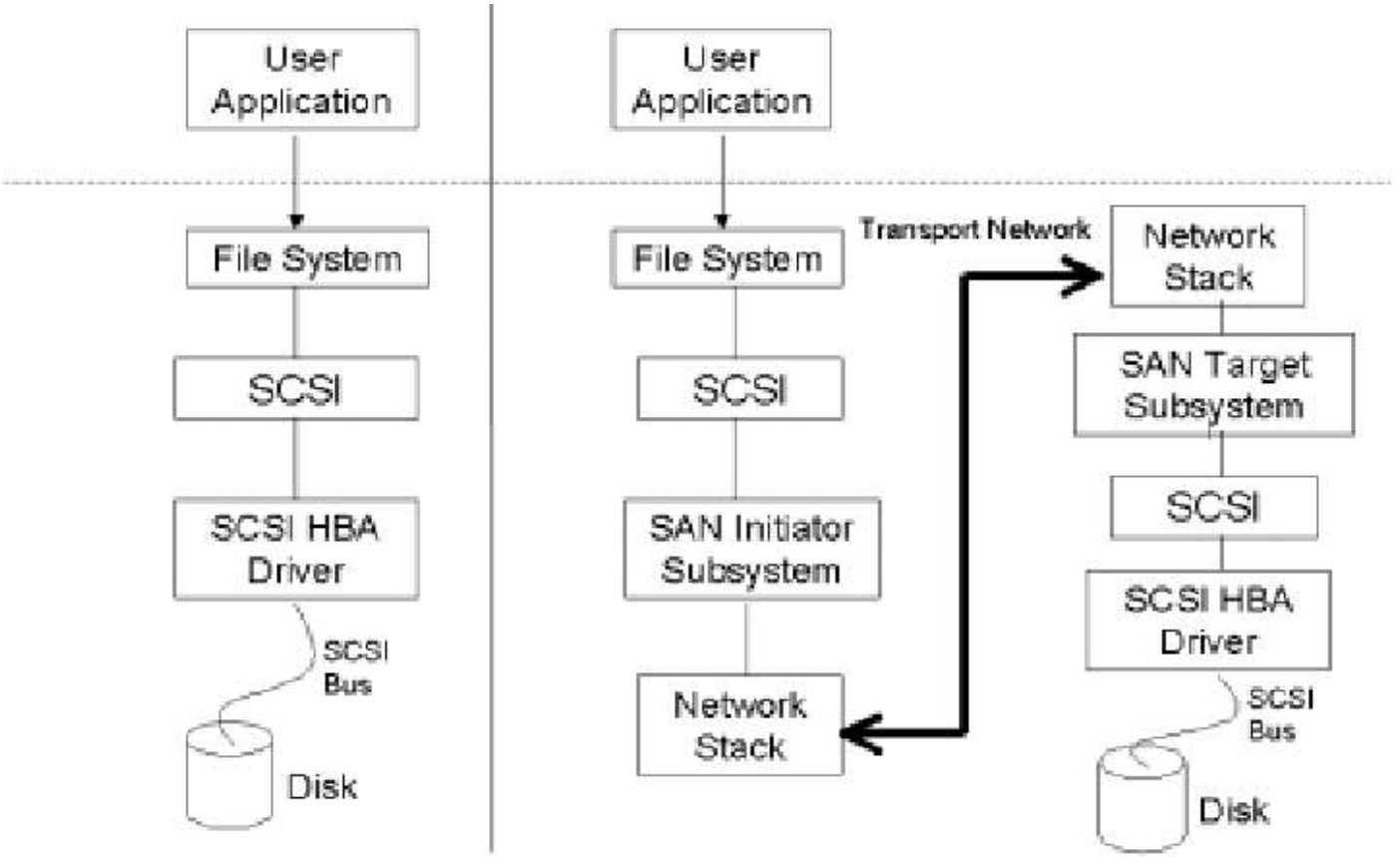
Sample SAN Configuration

Why do we need iSCSI?

- Limitation of SCSI
  - Point to point;
  - # of devices (8/16 or 7/15) and # of LUNs
  - Cable length; (25m in SCSI or 12m in Ultra SCSI)
- Limitation of FC-SAN
  - Interoperability issues and incompatibility with existing infrastructure;
  - Interoperability issues and incompatibility among different FC-SAN from different vendors;
  - Very expensive to install and maintain;

What is iSCSI?

- iSCSI – Internet SCSI protocol;
- A mapping of the SCSI over the TCP protocol.



Important Concepts in iSCSI

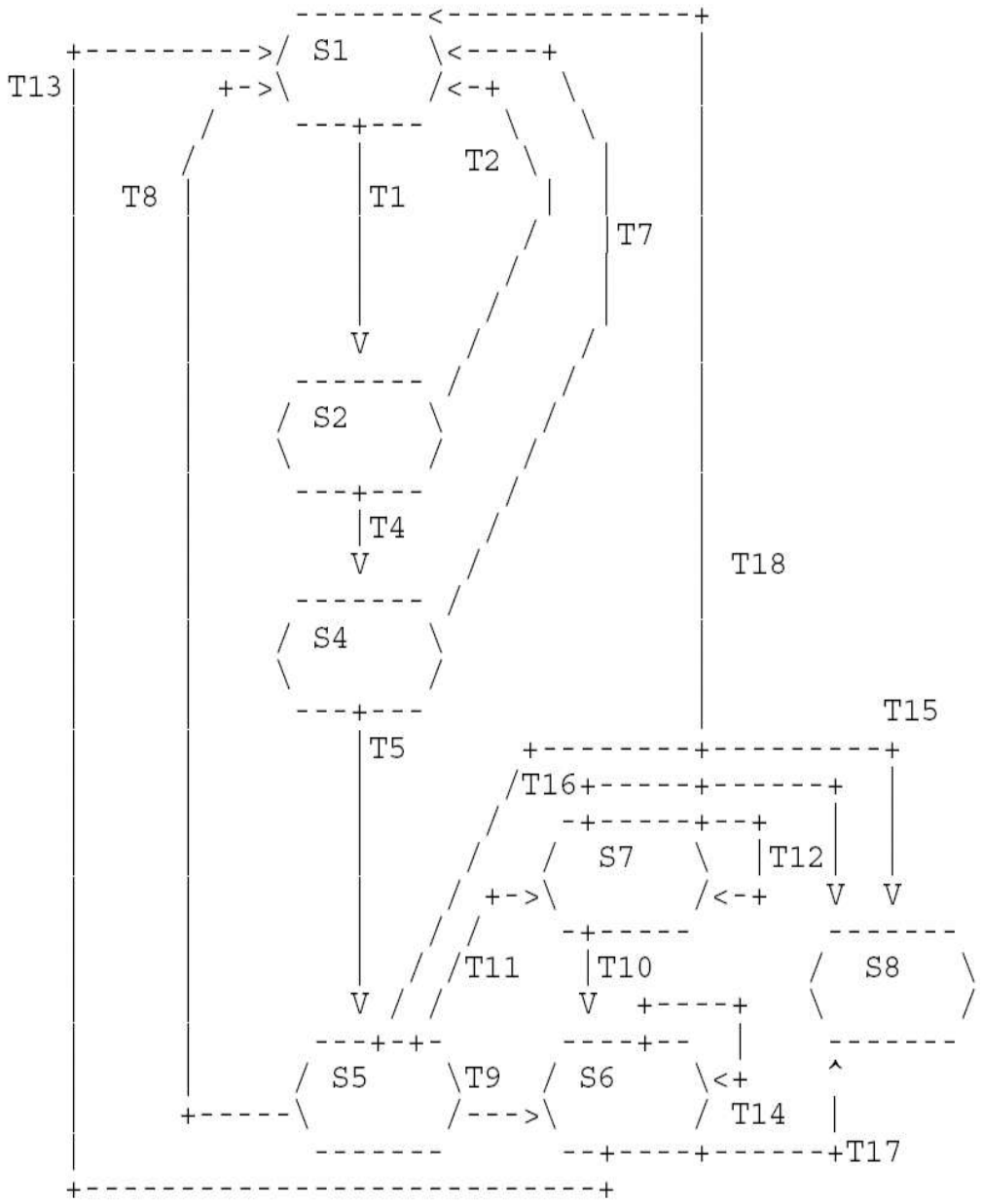
- ***Connection*** - A connection is a TCP connection. Communication and target occurs over one or more TCP connections. The TCP connections carry control messages, SCSI commands, parameters, and data within iSCSI Protocol Data Units (iSCSI PDUs).
- ***Session*** - The group of TCP connections that link an initiator with a session (loosely equivalent to a SCSI I-T nexus). TCP connections can be added and removed from a session. Across all connections within a session, an initiator sees one and the same target.
- ***Command*** - iSCSI Commands, SCSI Commands.
- ***CID***, ***SSID***, ***CmdSN***.

More about iSCSI Session/Connection

- Different session/connection phases
  - Login Phase
    - Creates a TCP connection;
    - Authenticates each party;
    - Negotiates operational parameters;
    - Creates or marks connection to a session;
  - Full Feature Phase
    - Data transfer
- Different session types:
  - Normal
  - Discovery

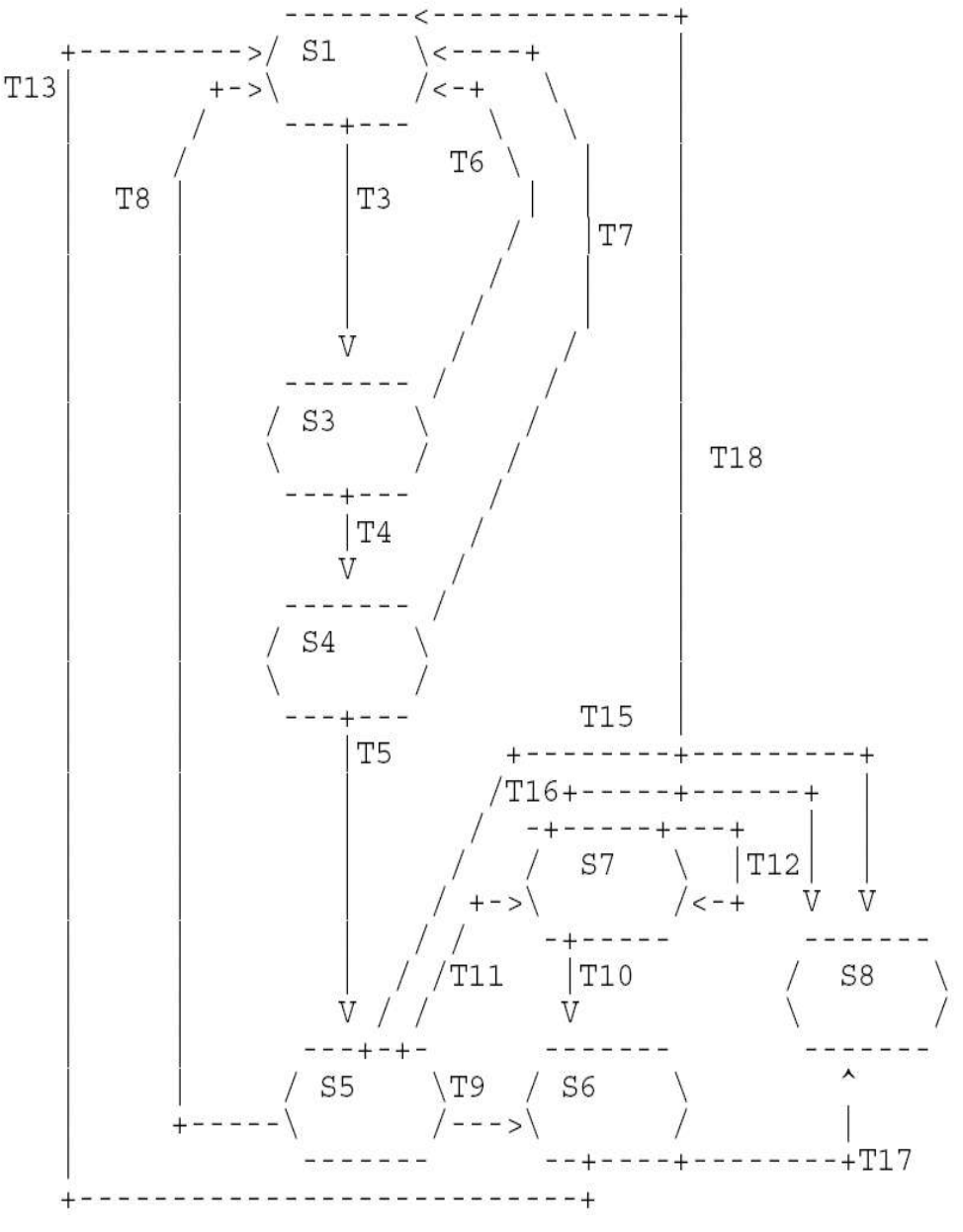
iSCSI Initiator Connection State Diagram

- S1: FREE
- S2: XPT_WAIT
- S4: IN_LOGIN
- **S5**: LOGGED_IN
- **S6**: IN_LOGOUT
- **S7**: LOGOUT_REQUESTE
- S8: CLEANUP_WAIT

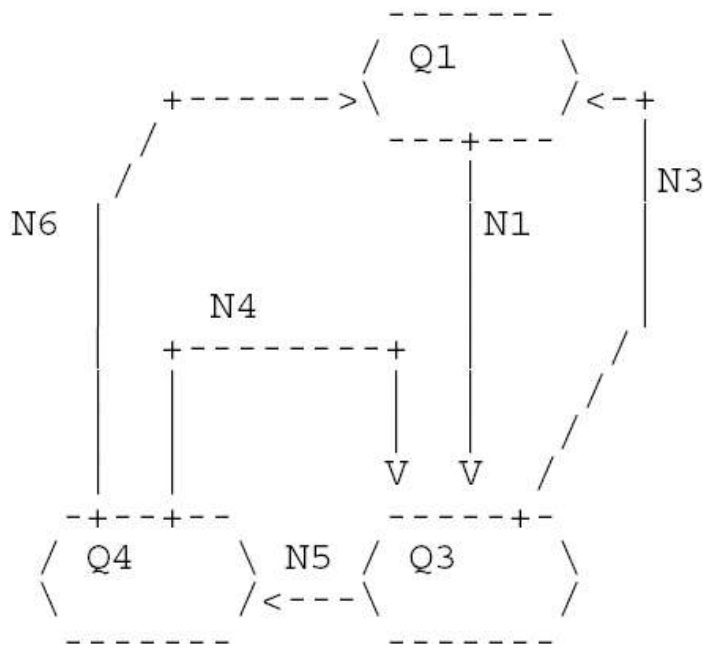


iSCSI Target Connection State Diagram

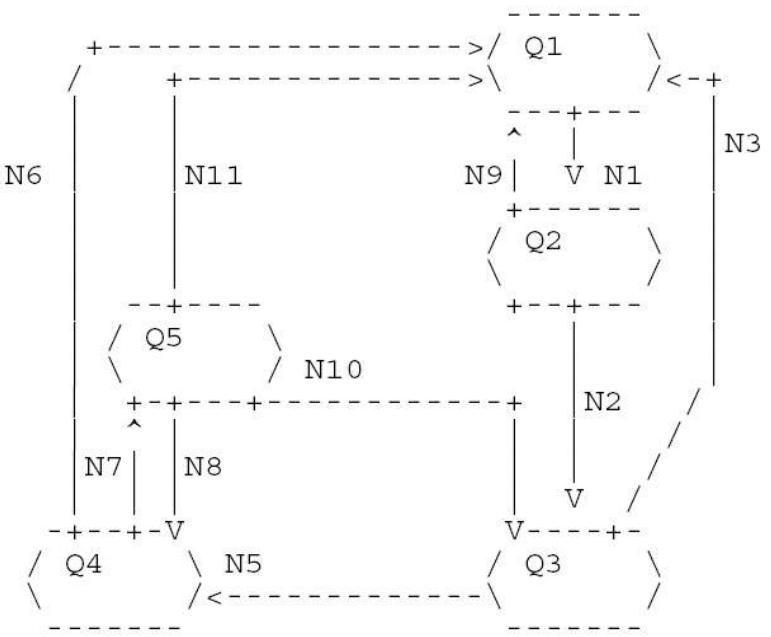
- S1: FREE
- S3: XPT_UP
- S4: IN_LOGIN
- **S5**: LOGGED_IN
- **S6**: IN_LOGOUT
- **S7**: LOGOUT_REQUESTED
- S8: CLEANUP_WAIT



iSCSI Session State Diagram



Initiator



Target

Q1: FREE, Q2: ACTIVE, **Q3**: LOGGED_IN,

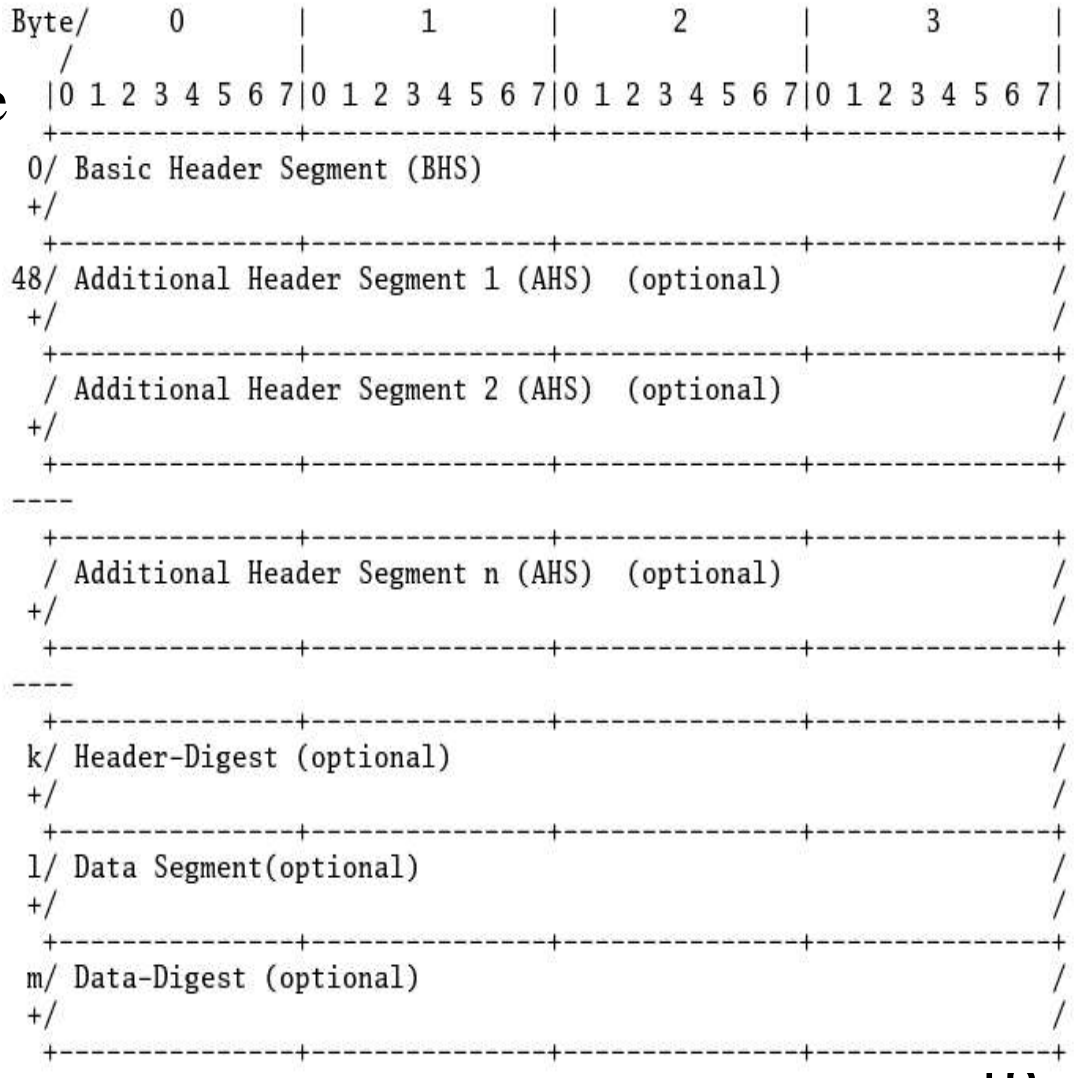
Q4: FAILED, Q5: IN_CONTINUE,

Negotiable Parameters

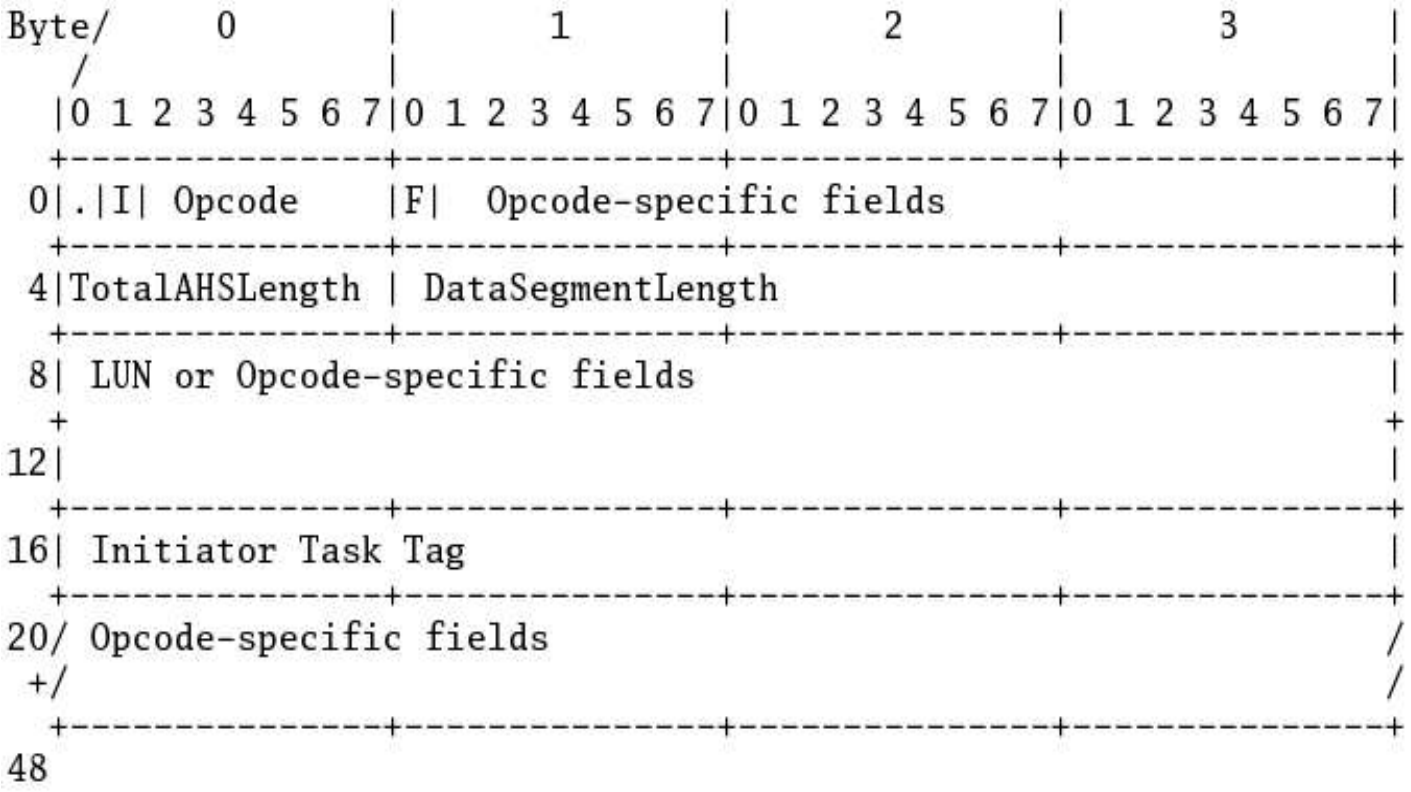
- Security parameters and operational parameters;
- “key=<value>” pairs;
- Some important operational parameters:
  - SessionType,
  - MaxConnections and MaxOutstandingR2T;
  - InitialR2T and ImmediateData;
  - MaxRecvDataSegmentLength, MaxRecvPDULength, DataPDULength,..

iSCSI PDU

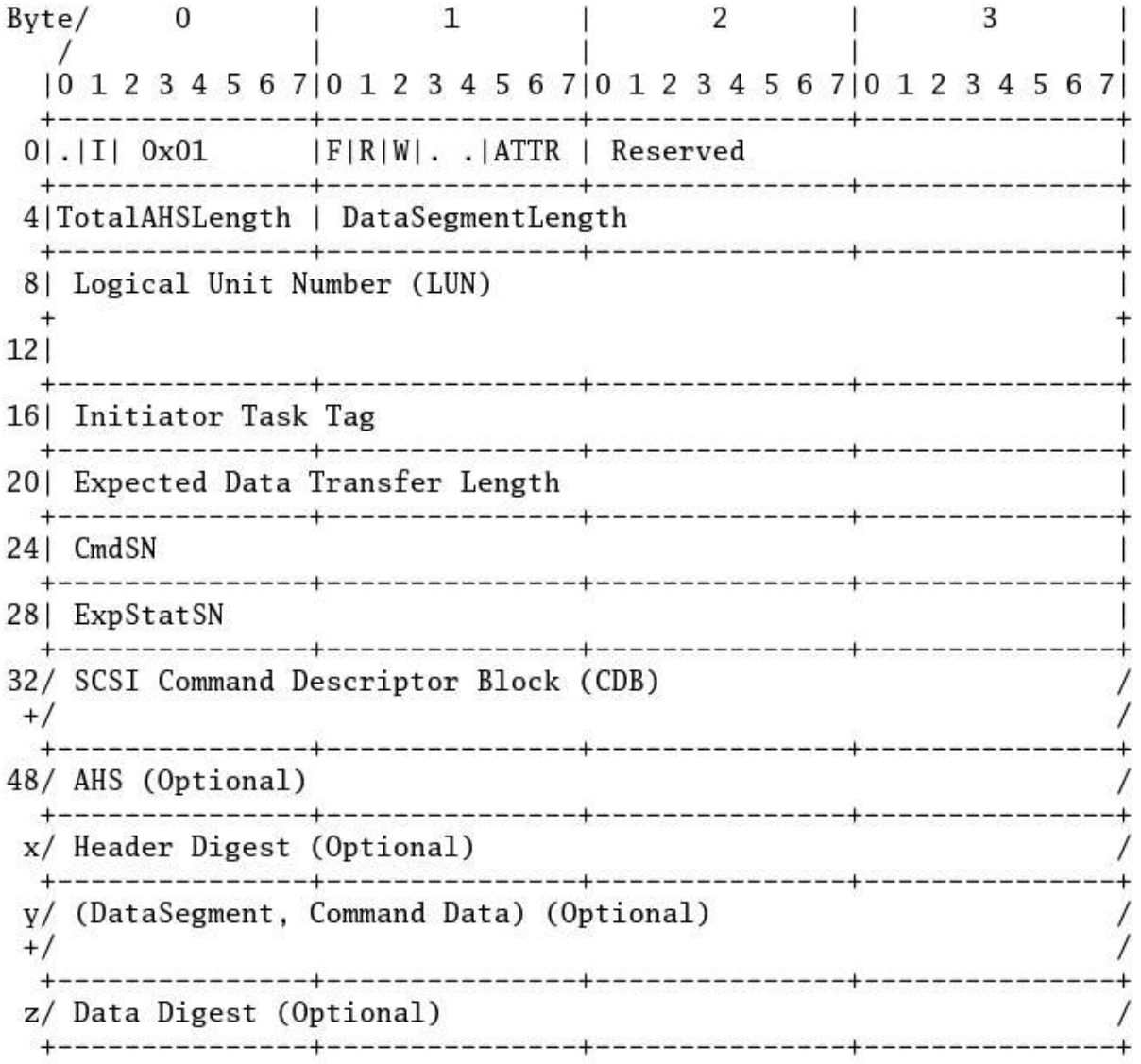
- PDU (Protocol Data Unit): The initiator and target divide their communications into messages. The term "iSCSI protocol data unit" (iSCSI PDU) is used for these messages.



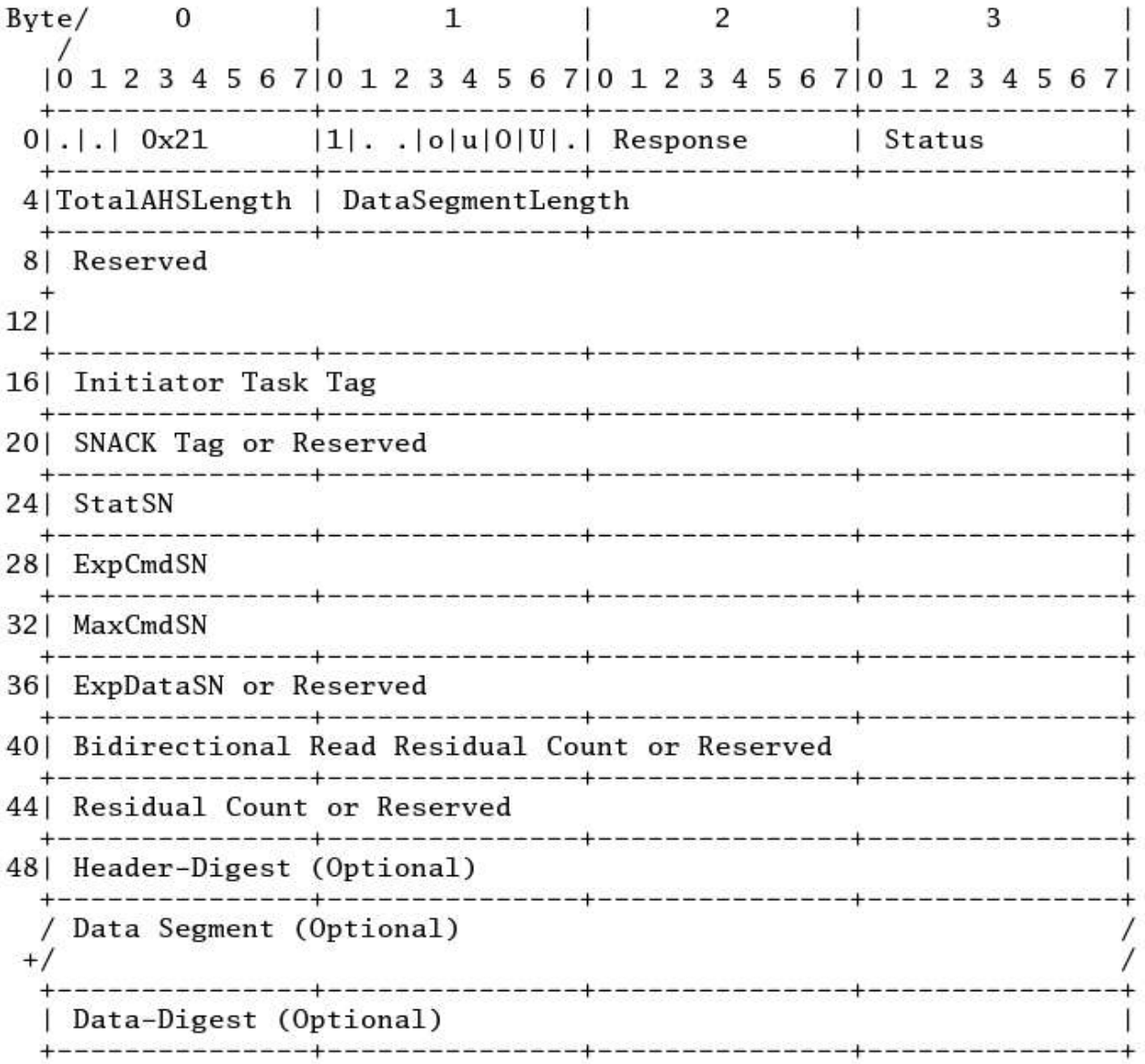
iSCSI Basic Header Segment



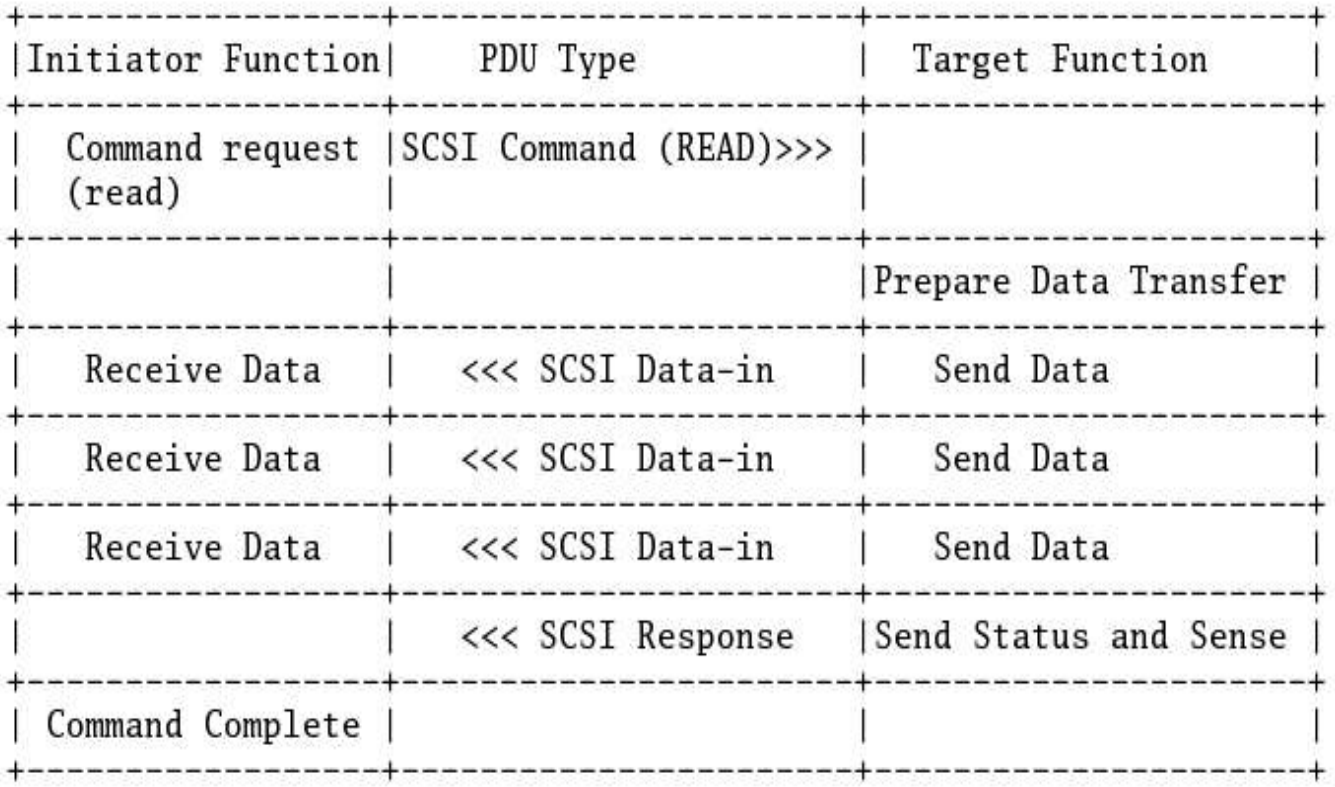
SCSI Command PDU



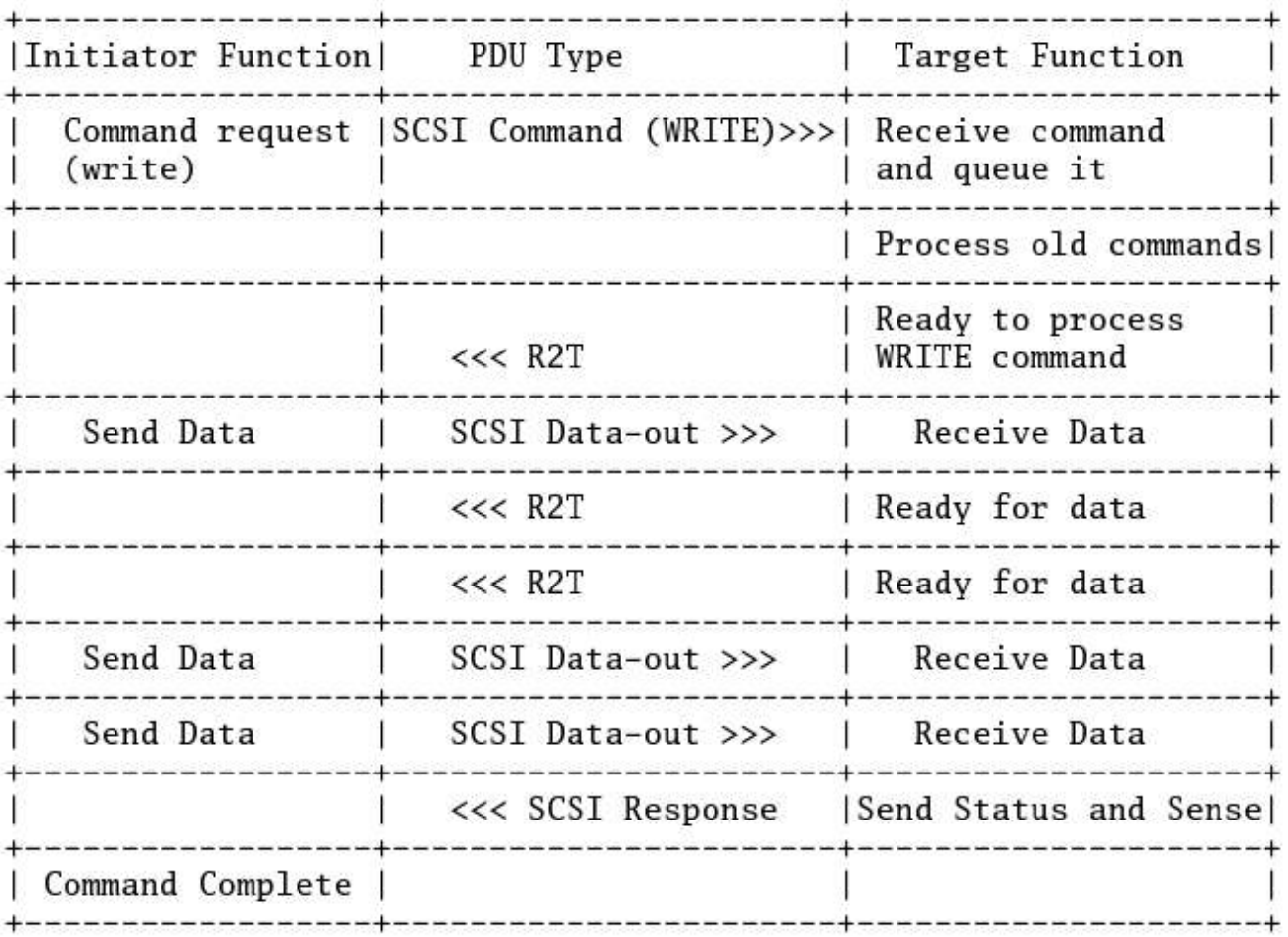
SCSI Response PDU



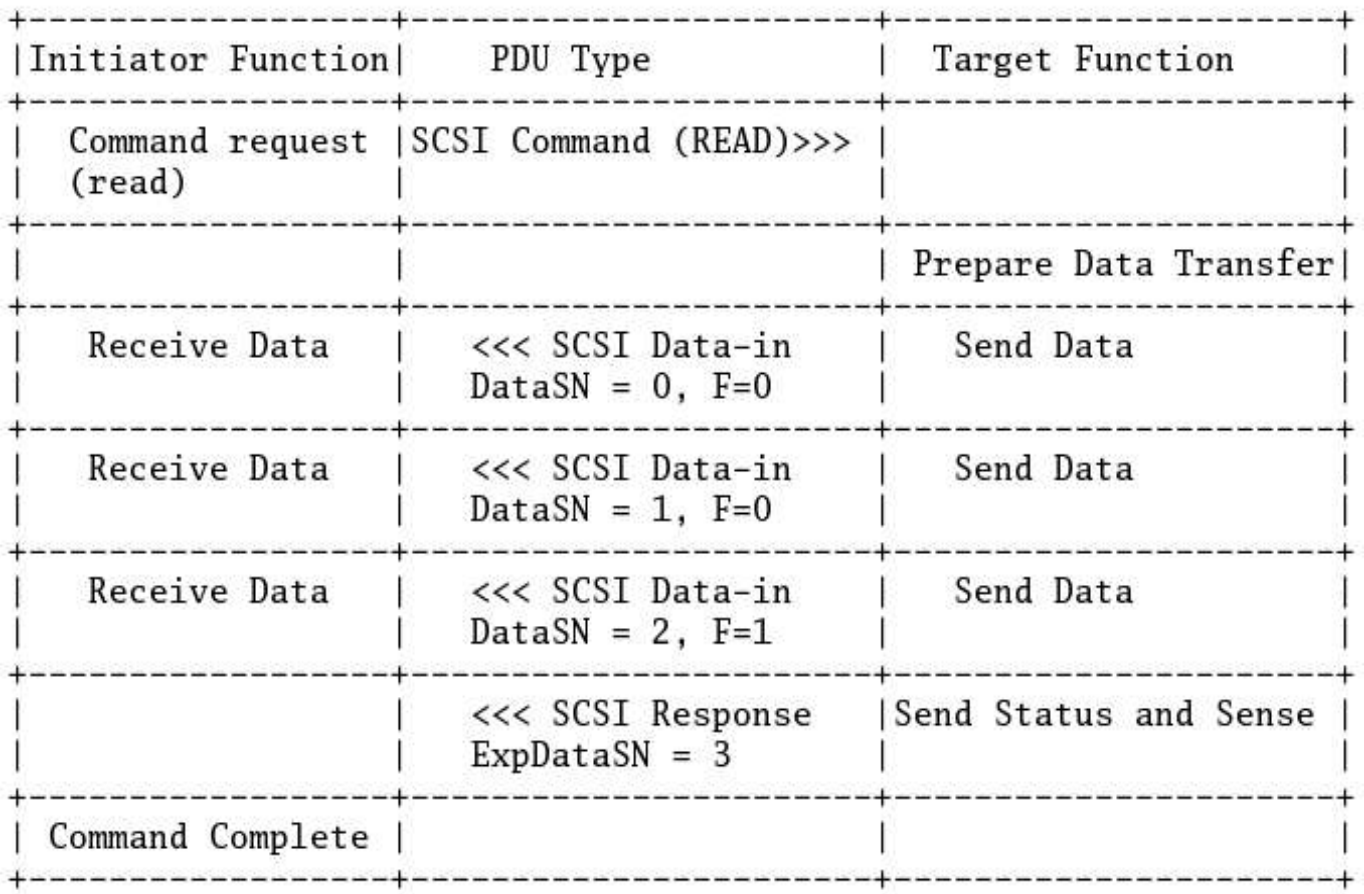
Simple Read


Initiator Function
PDU Type
Target Function
Command request (read)
SCSI Command (READ)>>>
Prepare Data Transfer
Receive Data
<<< SCSI Data-in
Send Data
Receive Data
<<< SCSI Data-in
Send Data
Receive Data
<<< SCSI Data-in
Send Data
<<< SCSI Response
Send Status and Sense
Command Complete

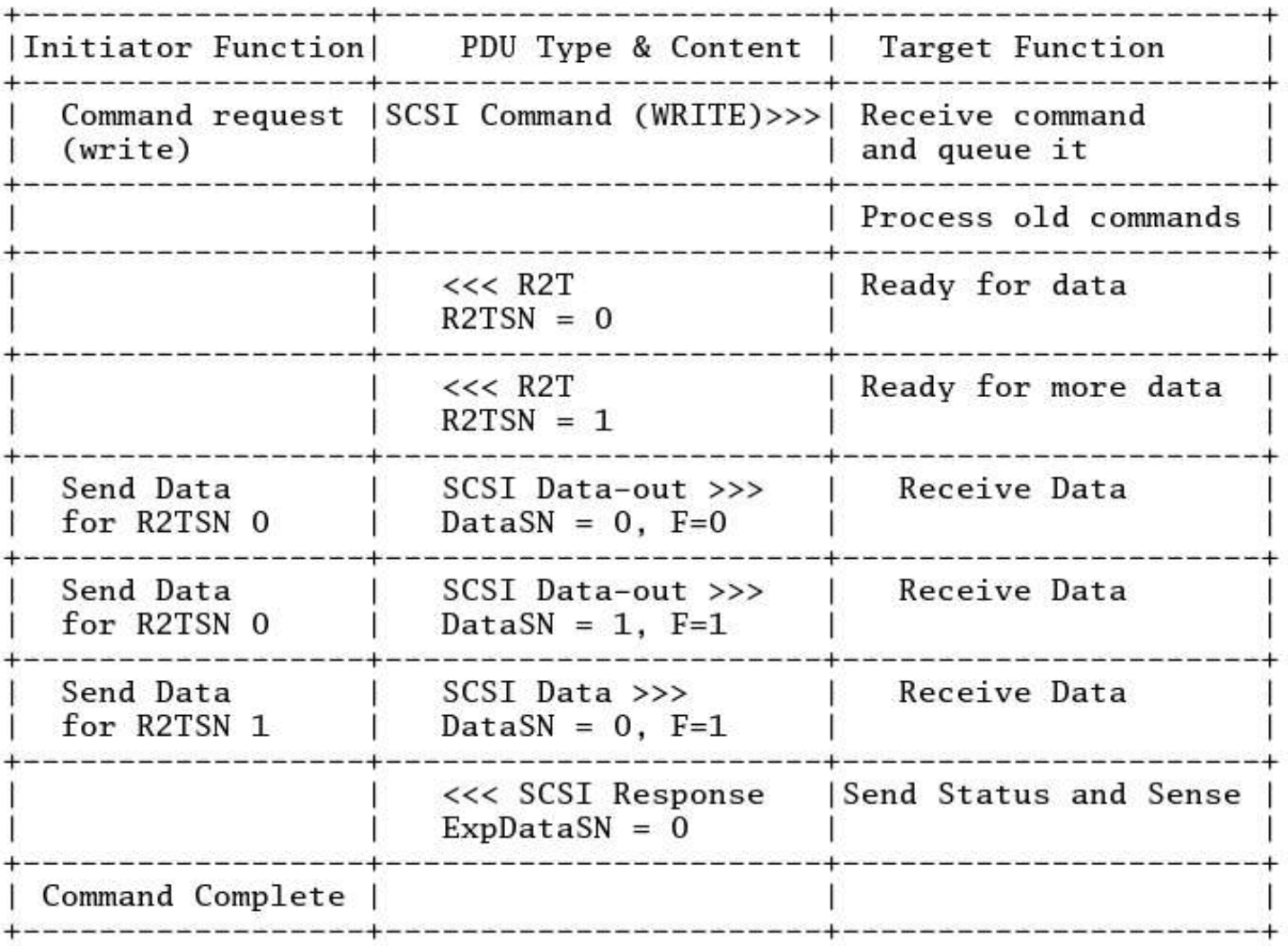
Simple Write



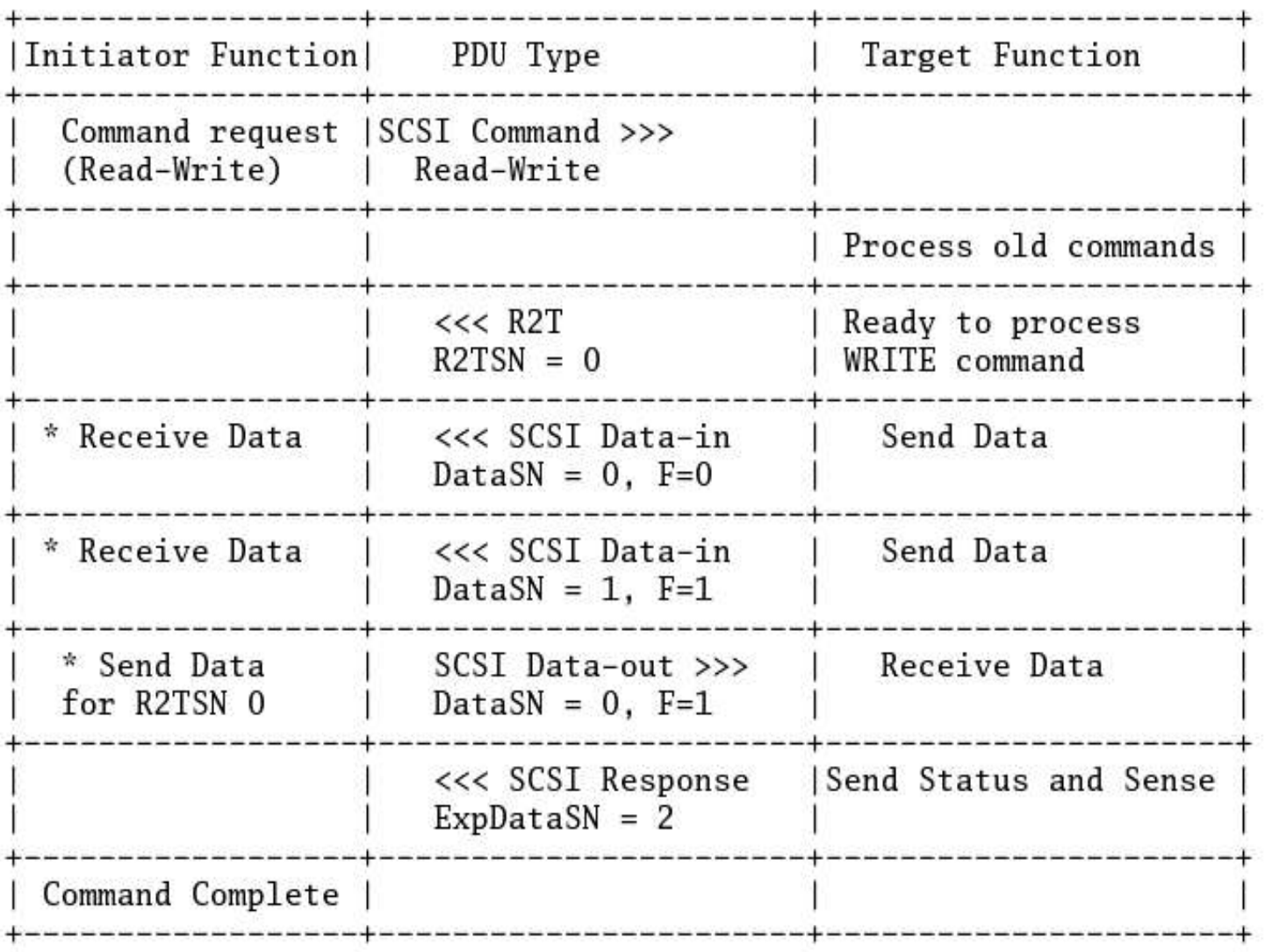
Read with DataSN



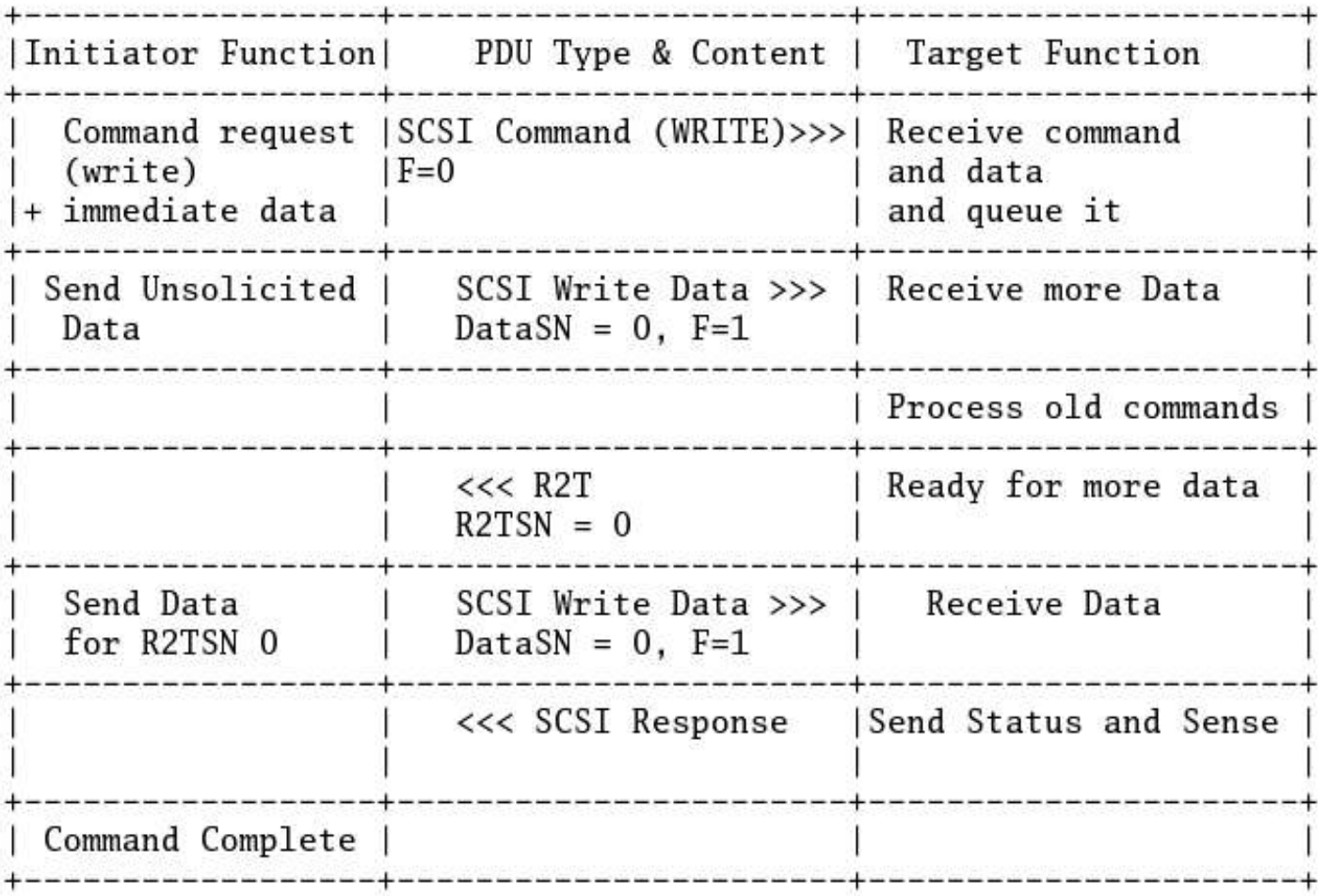
Write with DataSN



Bidirectional DataSN



Unsolicited and Immediate Write



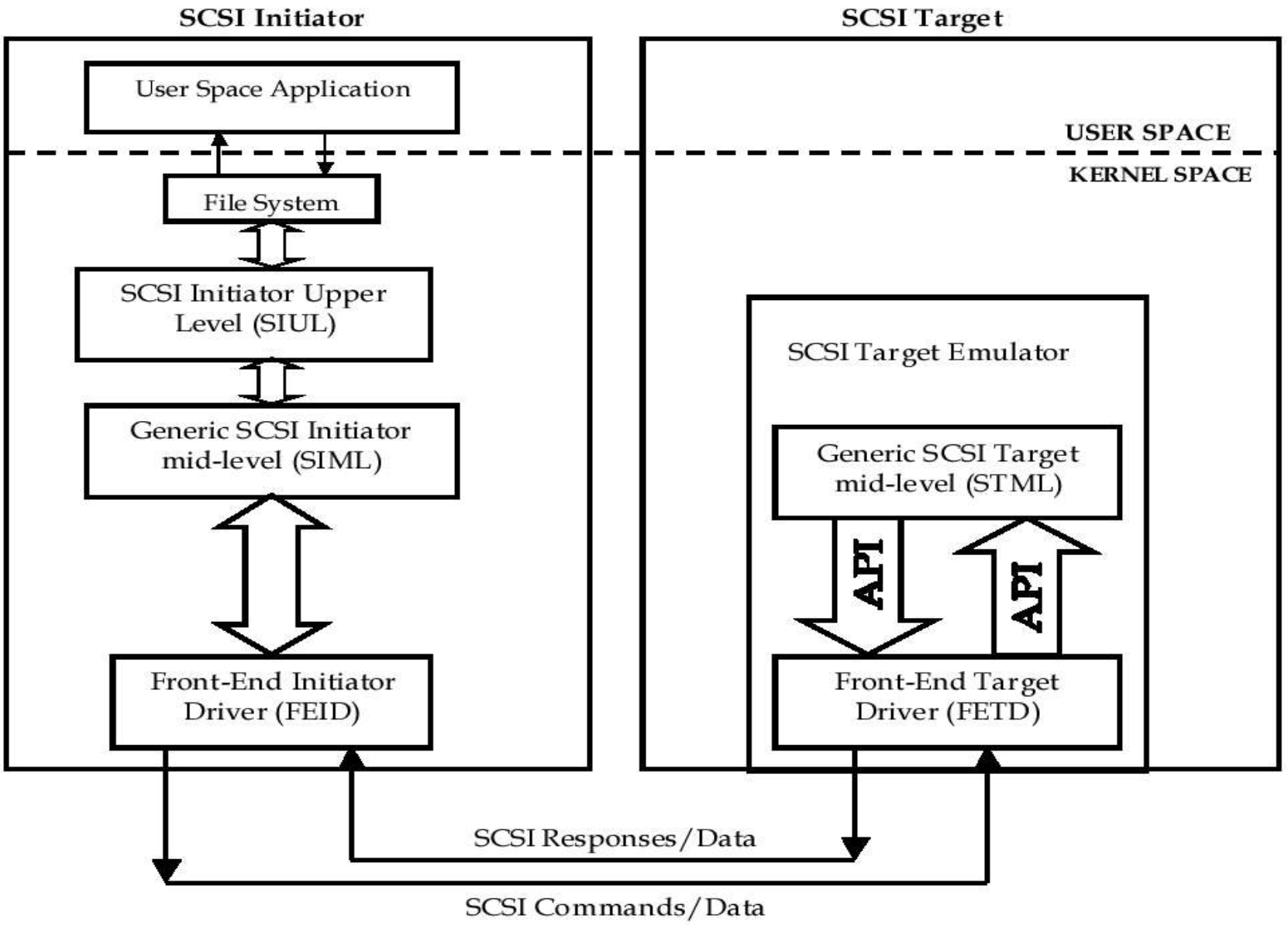
PDU Trace of a Read Request

- A READ request: 32768 bytes of data (64 * 512 bytes) starting LBA = 18572, ending LBA = 18635. Current initiator's CmdSN = 22264. Current target's StatSN counter is 67. MaxRecvPDULength = 12288, so we have 3 separate DataIn PDUs having DataSN numbers 0, 1, 2: the first 12288 bytes; the second 12288 bytes; the last 8192 bytes. DataPDUInOrder = yes.
- I->T: opcode = 0x01, F = 1, R = 1, DSL = 0, ITT = 88931, EDTL = 32768, CmdSN = 22264, ExpStatSN = 67, CDBopcode = 0x28, CDBlba = 18572, CDBlength = 64;
- T->I: opcode = 0x25, I = 1, F = 0, DSL = 12288, ITT = 88931, ExpCmdSN = 22265, DataSN = 0, BufferOffset = 0;
- T->I: opcode = 0x25, I = 1, F = 0, DSL = 12288, ITT = 88931, ExpCmdSN = 22265, DataSN = 1, BufferOffset = 12288;
- T->I: opcode = 0x25, I = 1, F = 0, DSL = 8192, ITT = 88931, ExpCmdSN = 22265, DataSN = 2, BufferOffset = 24576;
- T->I: opcode = 0x21, I = 1, F = **1**, Response = 0, Status = 0, DSL = 0, ITT = 88931, StatSN = 67, ExpCmdSN = 22265;

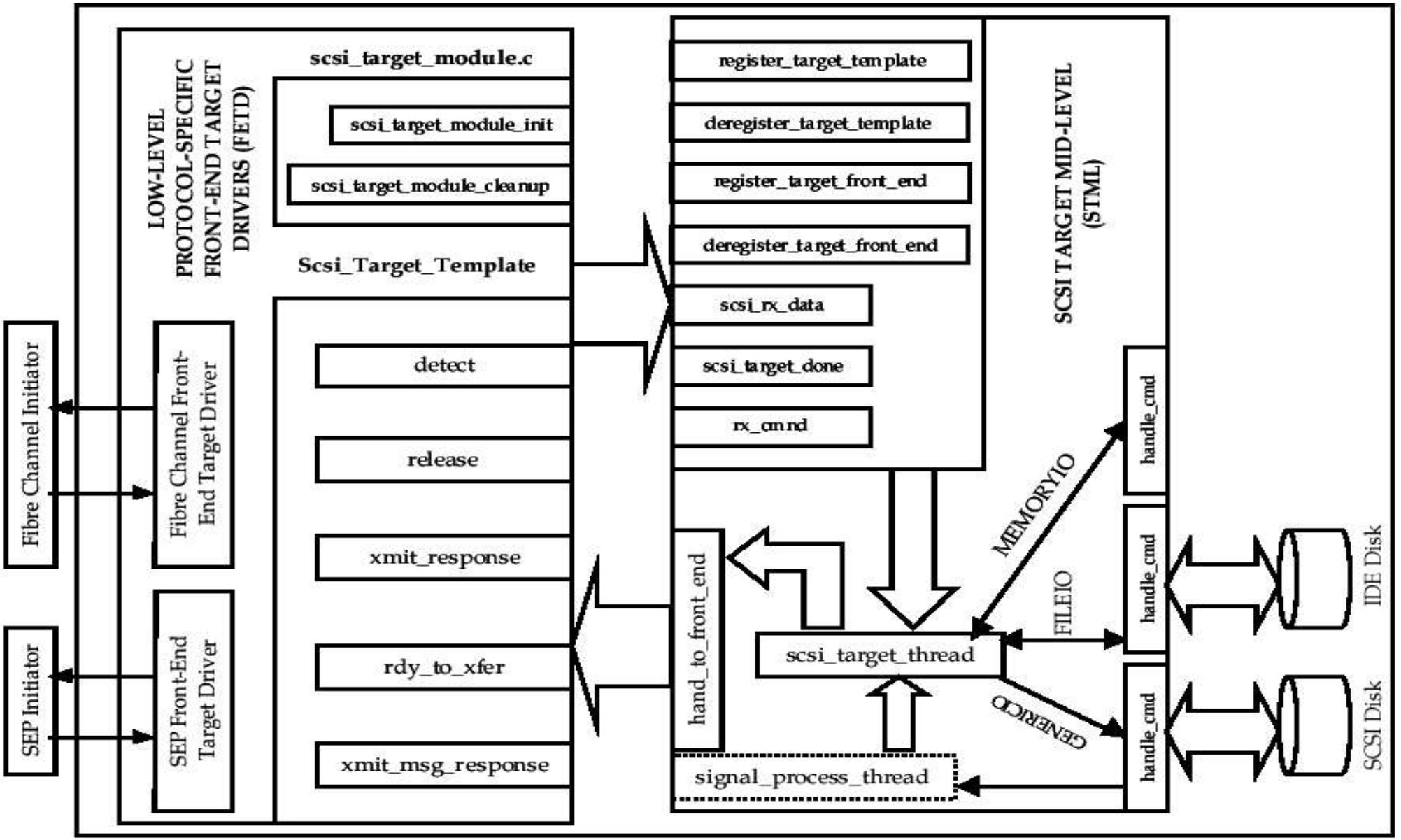
UNH iSCSI Reference Implementation

- Two-layer structure
  - SCSI middle layer (SIML, STML)
    - the common portions of what these Initiators and Targets need to do in terms of a logical unit of code that is responsible for processing SCSI commands, data and responses.
  - FED (FETD)
    - A relatively simpler front-end driver can be written that handles the details of the SCSI Transport Protocol itself.

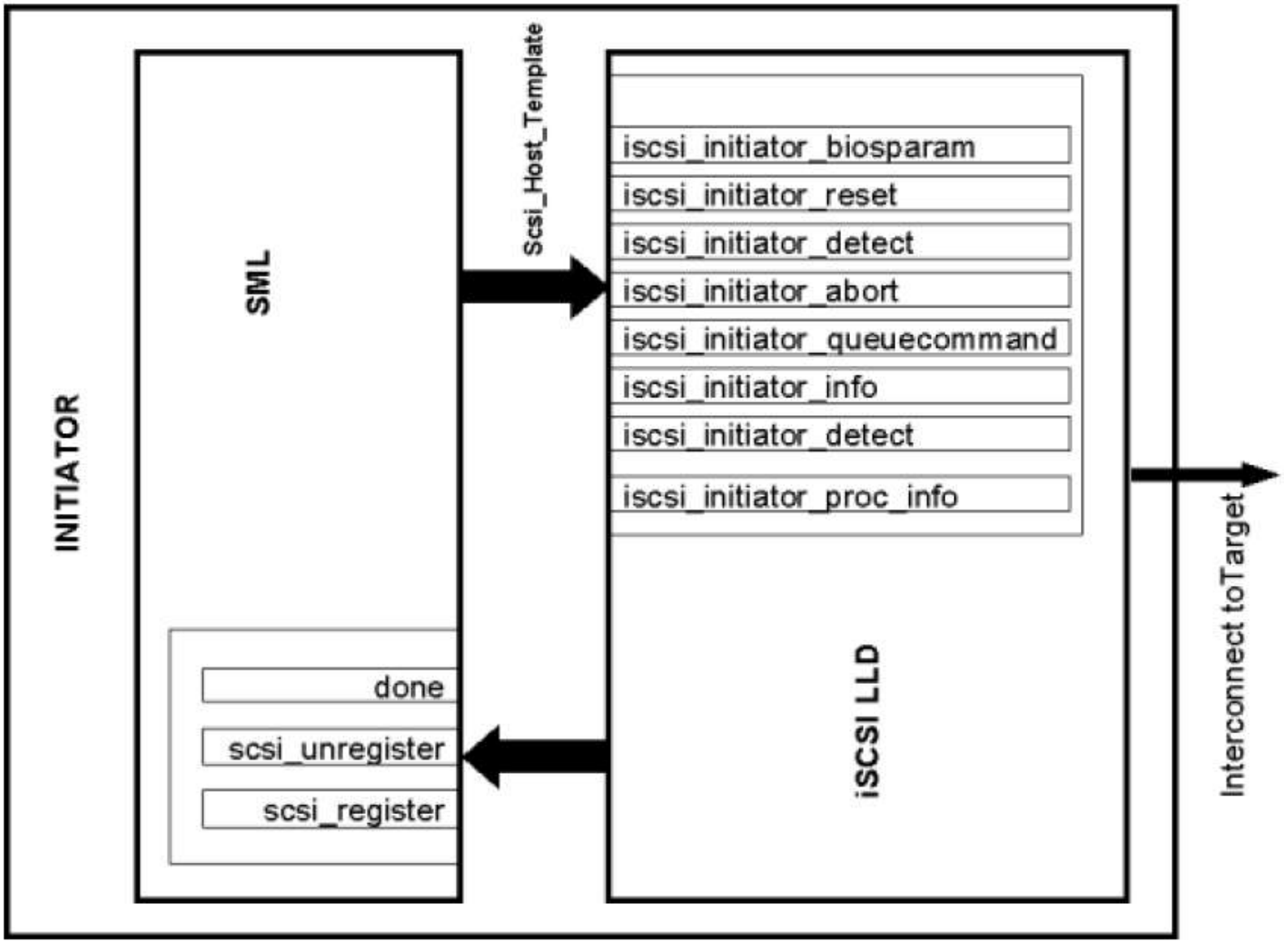
UNH Implementation Overview



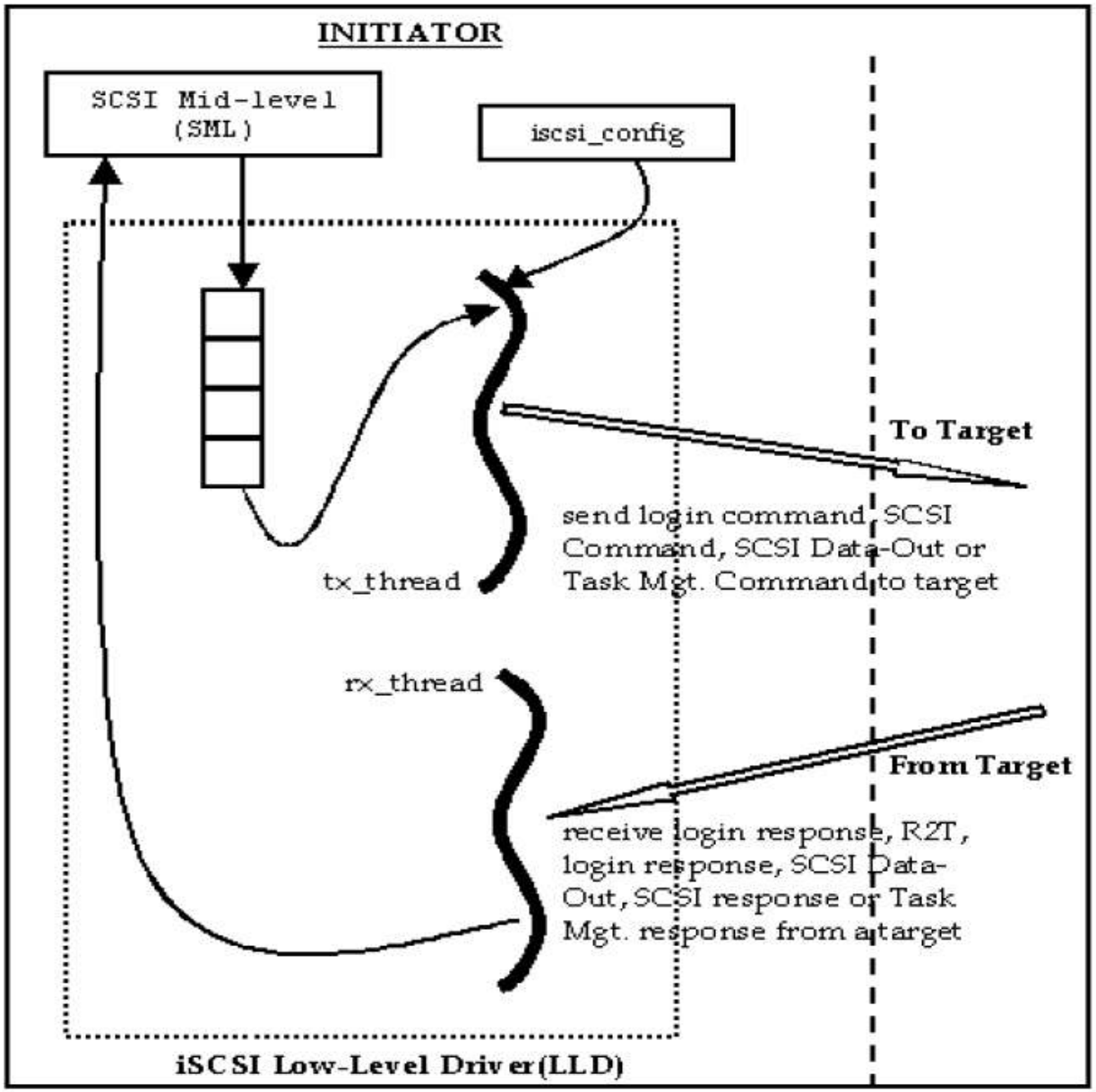
UNH Implementation Architecture



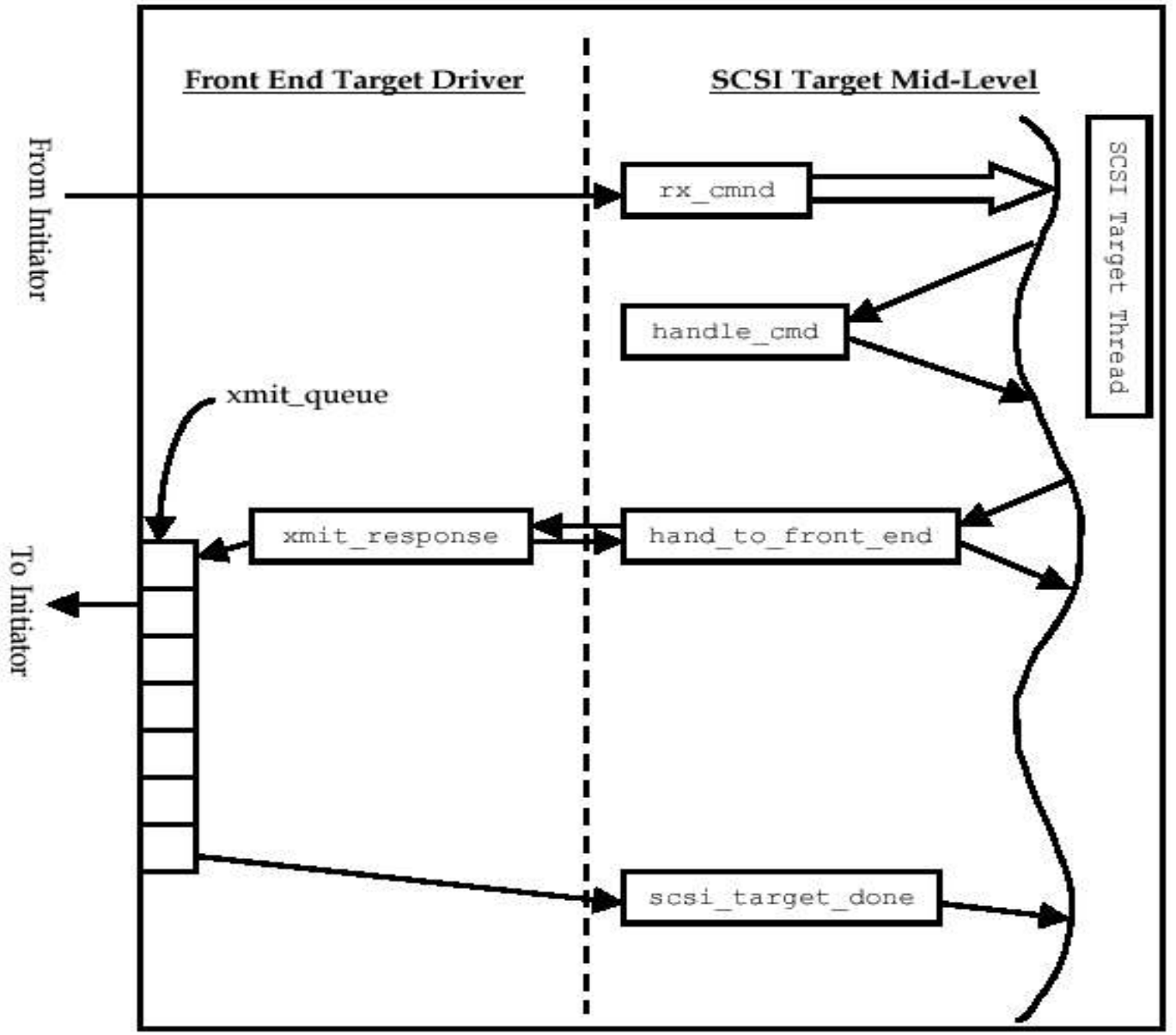
UNH Initiator Host Interface



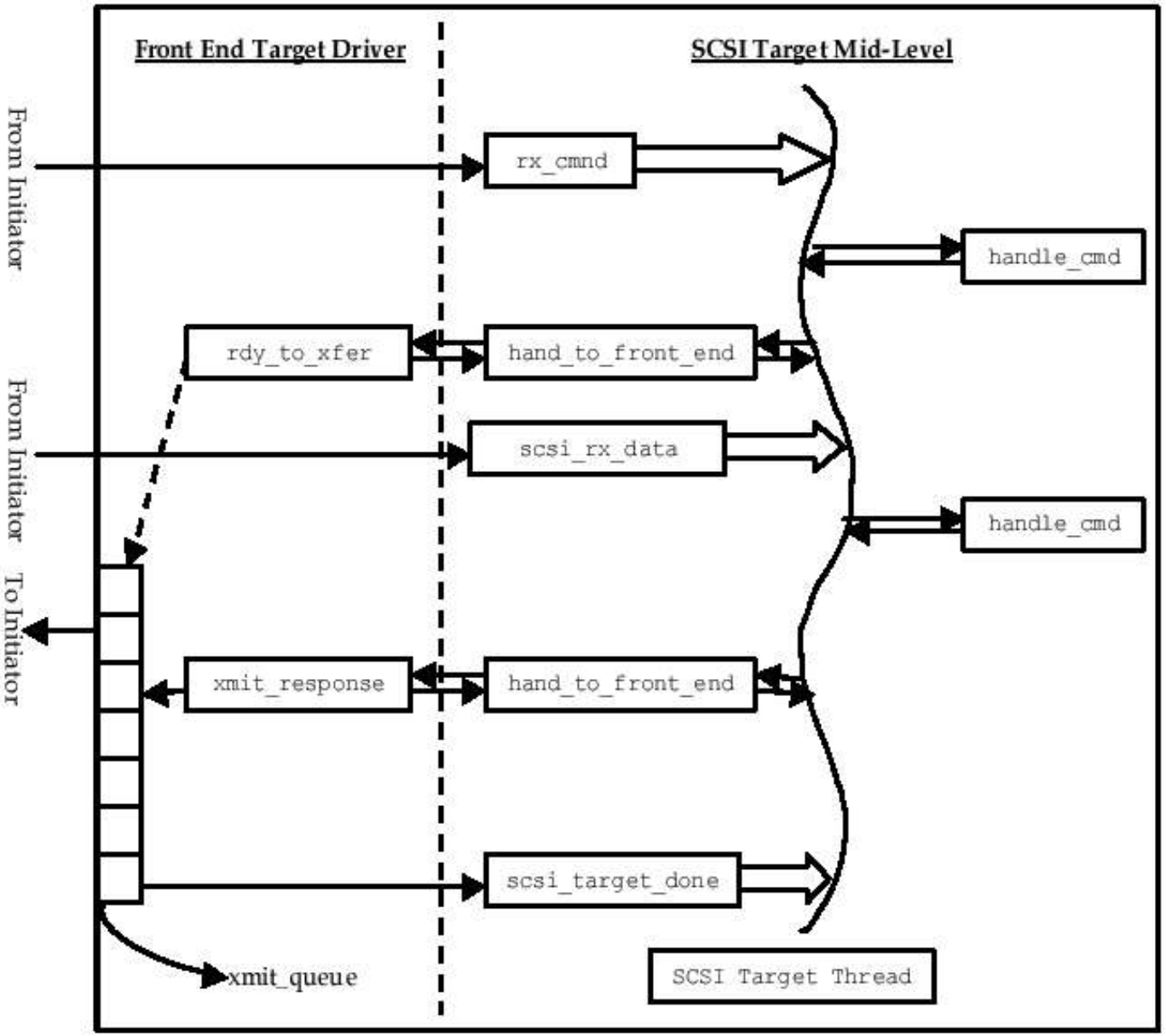
UNH Initiator Code



UNH Target Code Piece (READ)

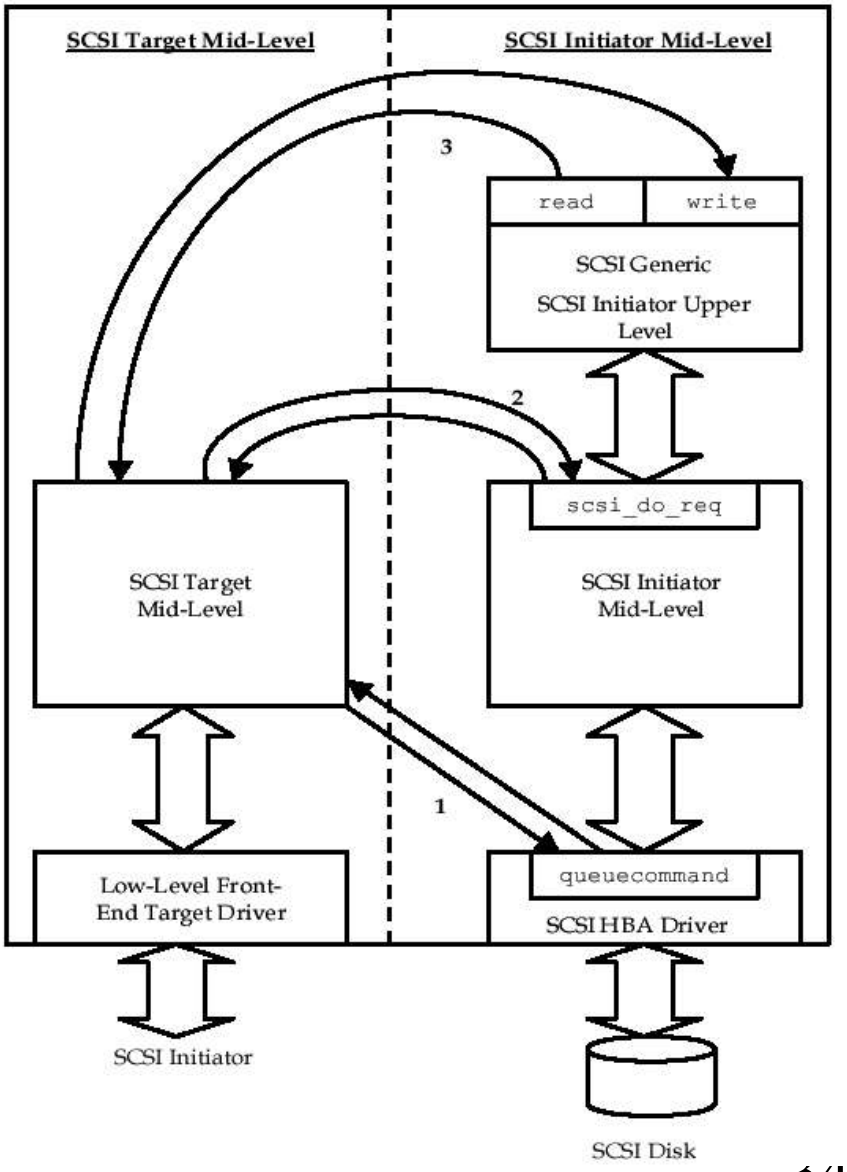


UNH Target Code Piece (WRITE)



UNH Target Storage Devices

- UNH code supports 3 storage modes: MEMORYIO, FILEIO, DISKIO;
- DISKIO mode can choose to operate on 3 different layers;



Limitation of iSCSI

- TCP/IP protocol overhead
  - Performance
  - Qos
- Network Latency
- Protocol overhead

Conclusion

- iSCSI is ...

Reference & More Information

- Most of the stuff in this presentation can be found from
  - iSCSI Standard:
    - http://www.haifa.il.ibm.com/satran/ips/draft-ietf-ips-iscsi-20.txt
  - University of New Hampshire Implementation:
    - Http://www.iol.unh.edu/consortiums/iscsi/
- Feel free to contact Ming Zhang (mingz@ele.uri.edu) if you have any question about the content in this presentation.